Interpretable Relevant Emotion Ranking with Event-Driven Attention

Yang Yang[†] **Deyu Zhou**[*†] **Yulan He**[§] **Meng Zhang**[†]
[†]School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, China
[§]Department of Computer Science, University of Warwick, UK
{yyang, d.zhou}@seu.edu.cn, y.he@cantab, m.zhang@seu.edu.cn.

Abstract

Multiple emotions with different intensities are often evoked by events described in documents. Oftentimes, such event information is hidden and needs to be discovered from texts. Unveiling the hidden event information can help to understand how the emotions are evoked and provide explainable results. However, existing studies often ignore the latent event information. In this paper, we proposed a novel interpretable relevant emotion ranking model with the event information incorporated into a deep learning architecture using the event-driven attentions. Moreover, corpus-level event embeddings and document-level event distributions are introduced respectively to consider the global events in corpus and the document-specific events simultaneously. Experimental results on three real-world corpora show that the proposed approach performs remarkably better than the state-of-the-art emotion detection approaches and multi-label approaches. Moreover, interpretable results can be obtained to shed light on the events which trigger certain emotions.

1 Introduction

The advent and prosperity of social media enable users to share their opinions, feelings and attitudes online. Apart from directly expressing their opinions on social media posts, users can also vote for their emotional states after reading an article online. An example of a news article crawled from Sina News Society Channel together with its associated emotion votes received from readers is illustrated in Figure 1. Treating these emotion votes as labels for the news article, we can define the emotion detection problem as an emotion ranking problem that ranks emotions based on their intensities. Moreover, some of the emotion labels

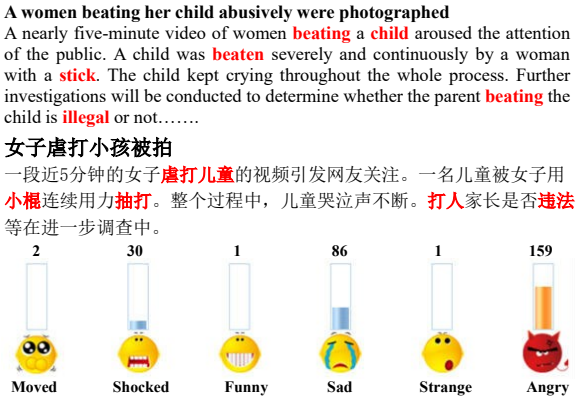

Figure 1: An example of an online news article with the readers' votes on various emotion categories. Words highlighted in red are event-indicative words.

could be considered as irrelevant emotions. For example, the emotion categories '*Moved*', '*Funny*' and '*Strange*' in Figure 1 only received one or two votes. These emotion votes could be noises (e.g., readers accidentally clicked on a wrong emotion button) and hence can be considered as irrelevant emotions. We need to separate the relevant emotions from irrelevant ones and only predict the ranking results for the relevant emotion labels. Therefore, the task we need to perform is the relevant emotion ranking. Understanding and automatically ranking users' emotional states would be potentially useful for downstream applications such as dialogue systems (Picard and Picard, 1997). Multiple emotion detection from texts has been previously addressed in (Zhou et al., 2016) which predicted multiple emotions with different intensities based on emotion distribution learning. A relevant emotion ranking framework was proposed in (Yang et al., 2018) to predict multiple relevant emotions as well as the rankings based on their intensities. However, existing emotion detection approaches do not model the events in texts

*Corresponding author

which are crucial for emotion detection. Moreover, most of the existing approaches only produce emotion classification or ranking results, and they do not provide interpretations such as identifying which event triggers a certain emotion.

We argue that emotions may be evoked by latent events in texts. Let us refer back to the example shown in Figure 1 and read the text more carefully. We notice that words such as '*beat*', '*child*' and '*stick*' marked in red are event-related words indicating the event of "*child abuse*" which may evoke the emotions of "*Anger*", "*Sadness*" and "*Shock*".

The above example shows that it is important to simultaneously consider the latent events in texts for relevant emotion ranking. In this paper we proposed an interpretable relevant emotion ranking model with event-driven attention (IRER-EA). We focus on relevant emotion ranking (RER) by discriminating relevant emotions from irrelevant ones and only learn the rankings of the relevant emotions based on their intensities.

Our main contributions are summarized below:

- A novel interpretable relevant emotion ranking model with event-driven attention (IRER-EA) is proposed. The latent event information is incorporated into a deep learning architecture through event-driven attentions which can provide clues of how the emotions are evoked with interpretable results. To the best of our knowledge, it is the first deep event-driven neural approach for RER.

- To consider event information comprehensively, corpus-level event embeddings are incorporated to consider global events in corpus and document-level event distributions are incorporated to learn document-specific event-related attention respectively.

- Experimental results on three different real-world corpora show that the proposed method performs better than the state-of-the-art emotion detection methods and multi-label learning methods. Moreover, the event-driven attention enables dynamically highlighting important event-related parts evoking the emotions in texts.

2 Related Work

In general, emotion detection methods can mainly be categorized into two classes: lexicon-based methods and learning-based methods. Lexicon-based approaches utilize emotion lexicons including emotion words and their emotion labels for detecting emotions from texts. For example, emotion lexicons are used in (Aman and Szpakowicz, 2007) to distinguish emotional and non-emotional sentences. Emotion dictionaries could also be used to predict the readers' emotion of new articles (Rao et al., 2012; Lei et al., 2014). Wang et al. (2015) proposed a model with several constraints using non-negative matrix factorization based on emotion lexicon for multiple emotion detection. However, these approaches often suffer from low recall.

Learning-based approaches can be further categorized into unsupervised and supervised learning methods. Unsupervised learning approaches do not require labeled training data (Blei et al., 2003). Supervised learning methods typically frame emotion detection as a classification problem by training supervised classifiers from texts with emotion categories (Rao et al., 2014; Wang and Pal, 2015; Rao, 2016). Lin et al. (2008) studied the readers' emotion detection with various kinds of feature sets on news articles. Quan et al. (2015) detected emotions from texts with a logistic regression model introducing the intermediate hidden variables to model the latent structure of input text corpora. Zhou et al. (2016) predicted multiple emotions with intensities based on emotion distribution learning. A relevant label ranking framework for emotion detection was proposed to predict multiple relevant emotions as well as the rankings of emotions based on their intensities (Yang et al., 2018; Zhou et al., 2018). However, these approaches do not model the latent events in texts.

In recent years, deep neural network models have been widely used for text classification. In particular, the attention-based recurrent neural networks (RNNs) (Schuster and Paliwal, 2002; Yang et al., 2016) prevail in text classification. However, these approaches ignore the latent events in texts thus fail to attend on event-related parts. Moreover, they are lack of interpretation.

Our work is partly inspired by (Yang et al., 2018) for relevant emotion ranking, but with the following significant differences: (1) our model incorporates corpus-level event embeddings and document-level event distributions by an event-driven attention mechanism attending to event-related words, which are ignored in the mod-

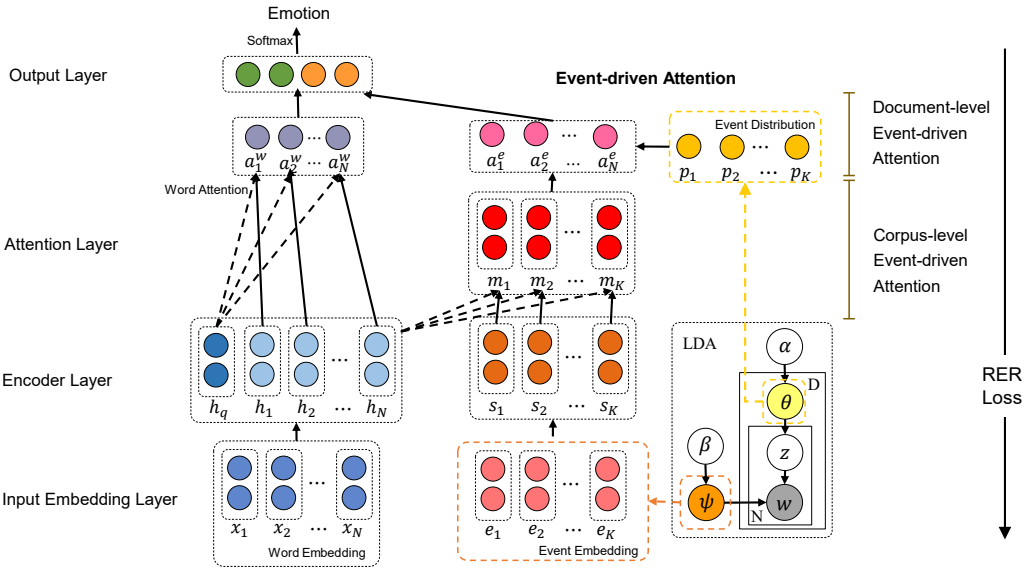

Figure 2: The IRER-EA model.

el (Yang et al., 2018) simply using a Kullback-Leibler (KL) divergence to approximatively learning the documents' topic distributions; (2) our model incorporates the event information into a deep learning architecture thus can consider the sequential information of texts which is ignored in the model (Yang et al., 2018) based on shallow bag-of-words representations.

3 The IRER-EA Model

3.1 Problem Setting

Assuming a set of T emotions, $L = \{l_1, l_2, ...l_T\}$, and a set of Q document instances, $\mathbb{D} = \{d_1, d_2, d_3, ..., d_Q\}$, each instance d_i is associated with a list of its relevant emotions $R_i \subseteq L$ ranked by their intensities and also a list of irrelevant emotions $\overline{R_i} = L - R_i$. Relevant emotion ranking aims to learn a score function $\mathbf{g}(d_i) = [g_1(d_i), ..., g_T(d_i)]$ which assigns a score $g_j(d_i)$ to each emotion $l_j, (j \in \{1, ..., T\})$. Relevant emotions and their rankings can be obtained simultaneously according to the scores assigned by the learned ranking function $\mathbf{g}$.

The learning objective of relevant emotion ranking (RER) is to both discriminate relevant emotions from irrelevant ones and to rank relevant emotions according to their intensities. Therefore, to fulfil the requirements of RER, the global objective function is defined as follows:

$$E = \sum_{i=1}^{Q} \sum_{l_t \in R_i} \sum_{l_s \in \prec(l_t)} \frac{1}{norm_{t,s}} [\exp(-(g_t(d_i) - g_s(d_i))) + \omega_{ts}(g_t(d_i) - g_s(d_i))^2] \quad (1)$$

Here, $l_s \in \prec (l_t)$ represents that emotion l_s is less relevant than emotion l_t. The normalization term $norm_{t,s}$ is used to avoid terms dominated by the sizes of emotion pairs. The term $g_t(d_i) - g_s(d_i)$ measures the difference between two emotions. ω_{ts} represents the relationship between two emotions l_t and l_s which is calculated by Pearson correlation coefficient (Nicewander, 1988).

We present the overall architecture of the proposed interpretable relevant emotion ranking with event-driven attention (IRER-EA) model in Figure 2. It consists of four layers: (1) the *Input Embedding Layer* including both word embeddings and event embeddings; (2) the *Encoder Layer* including both the word encoder and the event encoder; (3) the *Attention Layer* which computes the word-level attention scores and the event-driven attention scores taking into account of the corpus-level and document-level event information, respectively; (4) the *Output Layer* which generates the emotion ranking results.

3.2 Input Embedding Layer

The *Input Embedding Layer* contains word embeddings and event embeddings. Assuming a document d_i consisting of N words represented as

$d_i = \{w_1, w_2, ..., w_N\}$, the pre-trained word vector, GloVe (Pennington et al., 2014), is used to obtain the fixed word embedding of each word and d_i can be represented as $d_i = \{x_1, x_2, ..., x_N\}$ as shown in Figure 2.

Since nouns and verbs are more important than other word types in referring to specific events, they are utilized as inputs of topic model such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to generate events automatically. Therefore, the granularity of extracted events is controlled by the predefined K, the number of events. For the corpus $\mathbb{D}$ consisting of K events $\{e_1, e_2, ..., e_K\}$, the event embedding of the kth event e_k can be obtained from the output event-word distribution matrix E of the topic model. For the single document d_i, the event distribution $p = (p_1, p_2, ..., p_K)$ obtained from the topic model represents the probability of the text expressing each event.

3.3 Encoder Layer

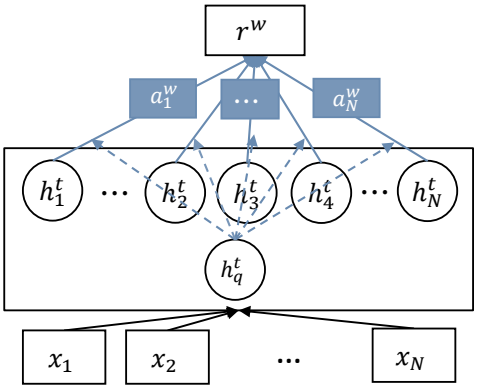

Figure 3: Word Encoder.

The *Encoder Layer* contains both the word encoder and event encoder. As for the word encoder, an alternative RNN structure (Zhang et al., 2018) is used to encode texts into semantic representations since it has been shown to be more effective in encoding longer texts. For document d_i, formally, a state at time step t can be denoted by:

$$H^t = < h_1^t, ..., h_N^t, h_q^t > \quad (2)$$

which consists of sub-states h_i^t for the ith word w_i in document d_i and a document-level sub-state h_q^t as shown in Figure 3. The hidden states are independent of each other at the present recurrent step and are connected across recurrent steps, which can capture long-range dependencies. The recurrent state transition process is used to model information exchange between those sub-states to enrich state representations incrementally. The state transition is similar to LSTM (Hochreiter and Schmidhuber, 1997) and a recurrent cell c_i^t for each word w_i and a cell c_q^t for document-level sub-state h_q are used. The value of each h_i^t is computed based on the values of x_i, h_{i-1}^{t-1}, h_i^{t-1}, h_{i+1}^{t-1}, h_q^{t-1} at two adjacent recurrent time steps. Note that the number of window size between two adjacent steps can be set manually. Hence, the hidden sub-states h_i for individual words w_i and a global document hidden state h_q for d_i are obtained.

As for event encoder, event representations are produced by the ReLU-actived neural perceptrons taking the event-word weight matrix $E \in V \times K$ as inputs. Hence, each event representation s_k representing event k is obtained according to the event embedding e_k, $k \in \{1, 2, 3, ..., K\}$.

3.4 Attention Layer

Given a word w_n in document d_i, h_n is the hidden representation of w_n after encoder. Given an event embedding e_k in the corpus, s_k is the event representation of e_k generated by the event encoder. Then we utilize attention weights to enhance the word representations and event representations from different aspects.

Our model contains two kinds of attention mechanisms including the word-level attentions and the event-driven attentions.

3.4.1 Word-Level Attention

As for word-level attentions, since not all words contribute equally to the meaning of a document, we introduce an attention mechanism to extract words with greater importance and aggregate the representations of those informative words to form the document representation, which is shown in the left part of Figure 2. More concretely,

$$\begin{aligned} \varphi_{w_i} &= tanh(W_w(h_i + h_q) + b_w) \\ a_i^w &= \frac{exp(\varphi_{w_i}^\top u_w)}{\sum_i exp(\varphi_{w_i}^\top u_w)} \\ r^w &= \sum_i a_i^w h_i \end{aligned} \quad (3)$$

where the weight a_i^w is the attention of the word w_i and W_w, b_w and u_w are parameters similar to (Pappas and Popescu-Belis, 2017). Note that we further incorporate the global information of the document representation h_q obtained from the encoder to strengthen the word attention.

3.4.2 Event-Driven Attention

In our model, we use the event-driven attention mechanism to attend to event-related words, which can discover words more important for text-related events. The event-driven attention leverages the corpus-level event information based on each event representation $s_k, k \in \{1, 2, 3, ..., K\}$ obtained from the corpus and the document-level event information based on the document's event distribution $p = (p_1, p_2, ..., p_K)$.

Corpus-level Event-Driven Attention

The model utilizes the corpus-level event information by a joint attention mechanism to consider global events in corpus, which aggregates the semantic representations $h = (h_1, h_2, ..., h_N)$ of an input text obtained and measures the interaction the words in the text with the event representations $s = (s_1, s_2, ..., s_K)$ by the event-driven attention. The corpus-level event-driven attention is calculated as follows:

$$\begin{aligned} \varphi^c &= tanh(W_c h + b_c) \\ m_k^c &= (\varphi^c)^\top s_k \\ a^c &= softmax(\sum_{k=1}^{K} m_k^c) \\ r^c &= (a^c)^\top h \end{aligned} \tag{4}$$

where $h = (h_1, h_2, ..., h_N)^\top$stands for the combination of all the hidden states of words in the document and W_c and b_c are parameters needed to be learnt for corpus-level event-driven attention. $\varphi^c = (\varphi_1^c, \varphi_2^c, ..., \varphi_N^c)$ refers to the hidden representation of state h through a fully connected layer. Given the event representation s_k, we measure the interaction of the words in the document and the event by an attention weight vector m_k^c which can be computed as the inner product of event s_k and φ^c followed by a softmax layer. $a^c = (a_1^c, a_2^c, ..., a_N^c)$ stands for the average attention weights of all the events for words which contribute to discover event keywords of a document according to different events in corpus. Then we construct the text representation r^c with the sum of hidden states weighted by a^c.

Document-level Event-driven Attention

We further incorporate the document-level event-driven attention mechanism. Our model can attend to the event distributions of the current document in order to strengthen the effect of the current document expressing each event and learn document-specific event related attention. For each document, $p = (p_1, p_2, ..., p_K)$ denotes the event distributions of the document, with each dimension representing the level of prominence of the corresponding event occurred in the document. The corpus-level event-driven attention weights can be further strengthened by including document-level event distributions. The document-level event-driven attention is calculated as follows:

$$\begin{aligned} \varphi^d &= tanh(W_d h + b_d) \\ m_k^d &= (\varphi^d)^\top s_k \\ a^e &= softmax(\sum_{k=1}^{K} m_k^d p_k) \\ r^e &= (a^e)^\top h \end{aligned} \tag{5}$$

where $h = (h_1, h_2, ..., h_N)^\top$ stands for the aggregation of all the hidden states of words in the document and W_d and b_d are parameters needed to be learnt for the document-level event-driven attention. $\varphi^d = (\varphi_1^d, \varphi_2^d, ..., \varphi_N^d)$ refers to the hidden representation of state h through a fully connected layer. m_k^d represents the interaction of the words in the document and the event which can be computed as the inner product of event s_k and φ^d. Then m_k^d is weighted by the document-level event distribution, $p = (p_1, p_2, ..., p_K)$, followed by a softmax layer, and $a^e = (a_1^e, a_2^e, ..., a_N^e)$ stands for the attention weight after incorporating the document-level event distributions. Then we construct the text representation r^e with the sum of hidden states weighted by a^e. Finally, r^e is used as the final text representation obtained by the event-driven attention which simultaneously takes into account both the corpus-level event information and the document-level event information.

3.5 Output Layer

At last, we concatenate both the representations calculated by the word-level attention and the event-driven attention to obtain the final representation $r = [r^w, r^e]$, which is fed to a multi-layer perceptron and a softmax layer for identifying relevant emotions and their rankings.

4 Experiments

To evaluate our proposed approach, we conducted experiments on the following three corpora:

Sina Social News (News) (Zhou et al., 2018) consists of 5,586 news articles collected from the Sina

news *Society* channel. Each document was kept together with the readers' emotion votes of the six emotions including *Funny*, *Moved*, *Angry*, *Sad*, *Strange*, and *Shocked*.

Ren-CECps corpus (Blogs) (Quan and Ren, 2010) is a Chinese data set containing 1,487 blogs annotated with eight basic emotions from writer's perspective, including *Anger*, *Anxiety*, *Expect*, *Hate*, *Joy*, *Love*, *Sorrow* and *Surprise*. The emotions are represented by their emotion scores in the range of $[0, 1]$. Higher scores represent higher emotion intensities.

SemEval (Strapparava and Mihalcea, 2007) contains 1,250 English news headlines extracted from Google news, CNN, and many other portals, which are manually annotated with a fine-grained valence scale of 0 to 100 across 6 emotions, including *Anger*, *Disgust*, *Fear*, *Joy*, *Sad* and *Surprise*.

News		Blogs		SemEval	
Category	#Votes	Category	#Scores	Category	#Scores
Touching	694,006	Joy	349.2	anger	12042
Shock	572,651	Hate	174.2	disgust	7634
Amusement	869,464	Love	610.6	fear	20306
Sadness	837,431	Sorrow	408.4	joy	23613
Curiosity	212,559	Anxiety	422.6	sad	24039
Anger	1,109,315	Surprise	59.2	surprise	21495
		Anger	116.4		
		Expect	385.5		
All	4,295,426	All	2526.1	All	109,129

Table 1: Statistics for the three corpora used in our experiments.

The statistics for the three corpora used in our experiments are shown in Table 1.

In our experiments, the News and Blog corpora were preprocessed using the python jieba segmenter[1] for word segmentation and filtering. The third corpus SemEval is in English and was tokenized by white space. Stanford CoreNLP[2] was applied for parts of speech tagging to obtain the nouns and verbs of the documents. Stop words and words appeared less than twice were removed from documents. We used the pre-trained Chinese GloVe and English GloVe[3] vectors as the word embeddings in the experiments and the dimension of the word embeddings was 300.

[1] https://github.com/fxsjy/jieba

[2] https://stanfordnlp.github.io/CoreNLP/index.html

[3] https://nlp.stanford.edu/projects/glove/

Name	Definition
PRO Loss	$\frac{1}{n}\sum_{i=1}^{n}\sum_{e_t\in R_i\cup\{\Theta\}}\sum_{e_s\in\prec(e_t)}\frac{1}{norm_{t,s}}l_{t,s}$ $l_{t,s}$ is a modified 0-1 error; $norm_{t,s}$is the set size of label pair(e_t, e_s)
Hamming Loss	$\frac{1}{nT}\sum_{i=1}^{n}\lvert\hat{R}_i\triangle R_i\rvert$ The predicted relevant emotions: $\hat{R}_i$.
Ranking Loss	$\frac{1}{n}\sum_{i=1}^{n}\frac{(\sum_{(e_t,e_s)\in R_i\times\overline{R_i}}\delta[g_t(x_i)<g_s(x_i)])}{(\lvert R_i\rvert\times\lvert\overline{R_i}\rvert)}$ where δ is the indicator function.
One Error	$\frac{1}{n}\sum_{i=1}^{n}\delta[\underset{e_t}{\operatorname{argmax}}\, g_t(x_i)\notin R_i]$
Average Precision	$\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\lvert R_i\rvert}\times\sum_{t:e_t\in R_i}\frac{\lvert\{e_s\in R_i\lvert g_s(x_i)>g_t(x_i)\}\rvert}{\lvert\{e_s\lvert g_s(x_i)>g_t(x_i)\}\rvert}$
Coverage	$\frac{1}{n}\sum_{i=1}^{n}\max_{t:e_t\in R_i}\lvert\{e_s\lvert g_s(x_i)>g_t(x_i)\}\rvert$
$F1_{exam}$	$\frac{1}{n}\sum_{i=1}^{n}\frac{2\lvert R_i\cap\hat{R}_i\rvert}{(\lvert R_i\rvert+\lvert\hat{R}_i\rvert)}$
MicroF1	$F1(\sum_{t=1}^{T}TP_t,\sum_{t=1}^{T}FP_t,\sum_{t=1}^{T}TN_t,\sum_{t=1}^{T}FN_t)$
MacroF1	$\frac{1}{T}\sum_{t=1}^{T}F1(TP_t,FP_t,TN_t,FN_t)$

Table 2: Evaluation criteria for the Multi-Label Learning (MLL) methods. TP_t, FP_t, TN_t, FN_t represent the number of true positive, false positive, true negative, and false negative test examples with respect to emotion t respectively. $F1(TP_t, FP_t, TN_t, FN_t)$ represent specific binary classification metric F1 (Manning et al., 2008).

The event embeddings and event distributions used in the proposed method are derived in different ways. For long documents including News and Blogs, LDA was employed to generate event embeddings and event distributions using verbs and nouns as the input. For short texts in SemEval with the sparsity problem, Bi-term Topic Model (BTM) (Cheng et al., 2014) was chosen. The number of topics was 60. The parameters were chosen from the validation set which is 10% of the training set. The encoder was trained using a learning rate of 0.001, a dropout rate of 0.5, a window size of 1 and a layer number of 3. The number of epochs was 10 and the mini batch (Cotter et al., 2011) size was 16. For each method, 10-fold cross validation was conducted to get the final results.

The baselines can be categorized into two classes, emotion detection methods and multi-label methods. Most these baselines are either re-implemented or cited from published papers. For instance, the results of multi-label methods are re-implemented, since they are not proposed for relevant emotion ranking. The performances of some emotion detection methods, such as EDL, EmoDetect, RER and INNRER, are cited from the pub-

Corpus	Category	Method	Criteria								
			PL(↓)	HL(↓)	RL(↓)	OE(↓)	AP(↑)	Cov(↓)	F1(↑)	MiF1(↑)	MaF1(↑)
News	Emotion Detection	EDL	0.2348	0.2510	0.1616	0.2243	0.8372	2.1940	0.6260	0.6454	0.5703
		EmoDetect	0.2157	0.2575	0.1538	0.1627	0.8605	2.1761	0.6697	0.6739	0.5359
		RER	0.2142	0.2498	0.1491	0.1513	0.8633	2.1989	0.6820	0.6919	0.6198
		INN-RER	0.1973	0.2312	0.1353	0.1331	0.8764	2.1339	0.7108	0.7161	0.6282
	Multi-label	LIFT	0.2224	0.3363	0.1382	0.1411	0.8234	2.1394	0.6646	0.6801	0.6151
		Rank-SVM	0.2842	0.2872	0.2114	0.2034	0.7967	2.5358	0.5066	0.5656	0.5298
		MLLOC	0.4458	0.4206	0.4500	0.4193	0.6531	3.9032	0.3000	0.4060	0.3327
		BP-MLL	0.2118	0.2399	0.1443	0.1544	0.8677	2.1738	0.6881	0.6915	0.6013
	Our model	**IRER-EA**	**0.1674**	**0.1958**	**0.0989**	**0.0846**	**0.9137**	**1.9108**	**0.7475**	**0.7379**	**0.6358**
Blogs	Emotion Detection	EDL	0.3385	0.3916	0.2550	0.4206	0.6962	4.2491	0.5060	0.5396	0.4131
		EmoDetect	0.3115	0.3848	0.2123	0.2880	0.7617	4.1650	0.5340	0.5492	0.4387
		RER	0.3007	0.3657	0.2043	0.2728	0.7746	4.1638	0.5957	0.6084	**0.5342**
		INN-RER	0.2829	0.3209	0.1924	0.2626	0.7784	3.6418	0.6187	0.6225	0.5133
	Multi-label	LIFT	0.3452	0.3817	0.3089	0.3306	0.7557	**3.1290**	0.6053	0.6113	0.5155
		Rank-SVM	0.3888	0.3786	0.3356	0.3219	0.7030	4.0801	0.3489	0.3686	0.3210
		MLLOC	0.4999	0.5135	0.5115	0.6554	0.5569	5.7432	0.4104	0.4411	0.4391
		BP-MLL	0.2987	0.3281	0.2141	0.2727	0.7267	3.9802	0.5844	0.6065	0.4833
	Our model	**IRER-EA**	**0.2609**	**0.2874**	**0.1638**	**0.1845**	**0.8149**	3.6510	**0.6304**	**0.6320**	0.4804
SemEval	Emotion Detection	EDL	0.4130	0.4291	0.3401	0.3875	0.7345	3.3433	0.4002	0.413	0.3813
		EmoDetect	0.3176	0.3167	0.2411	0.2308	0.8241	3.0439	0.6275	0.6245	0.5385
		RER	0.2907	0.3128	0.2389	0.2220	0.8302	2.9963	0.6839	0.6898	0.6283
		INN-RER	0.3194	**0.3005**	0.2302	0.2261	0.8379	**2.8632**	0.7081	0.7156	0.6093
	Multi-label	LIFT	0.4279	0.4651	0.3627	0.4113	0.7344	3.2823	0.6299	0.6469	0.6112
		Rank-SVM	0.3452	0.3617	0.3083	0.3006	0.7557	3.1290	0.6253	0.6472	0.5955
		MLLOC	0.4458	0.4206	0.4500	0.4193	0.6531	3.9032	0.3000	0.4060	0.3327
		BP-MLL	0.3790	0.3656	0.3605	0.3790	0.7945	3.2097	0.5868	0.6101	0.5402
	Our model	**IRER-EA**	**0.2754**	0.3065	**0.1999**	**0.1286**	**0.8976**	3.5799	**0.7394**	**0.7538**	**0.6841**

Table 3: Comparison with Emotion Detection Methods and Multi-label Methods. 'PL' represent Pro Loss, 'HL' represents Hamming Loss, 'RL' represents ranking loss, 'OE' represents one error, 'AP' represent average precision, 'Cov' represent coverage, 'F1' represents $F1_{exam}$, 'MiF1' represents MicroF1, 'MaF1' represents MacroF1. "↓" indicates "the smaller the better", while "↑" indicates "the larger the better". The best performance on each evaluation measure is highlighted by boldface.

lished paper (Yang et al., 2018) as they use the same experimental data as ours.

Evaluation metrics typically used in multi-label learning and label ranking are employed in our experiments which are different from those of classical single-label learning systems (Sebastiani, 2001). The detailed explanations of evaluation metrics are presented in Table 2.

4.1 Compared Methods

There are several emotion detection approaches addressing multiple emotions detection from texts.

- **Emotion Distribution Learning(EDL)** (Zhou et al., 2016) learns a mapping function from sentences to their emotion distributions.
- **EmoDetect** (Wang and Pal, 2015) employs a constraint optimization framework with several constraints to obtain multiple emotions.
- **RER** (Zhou et al., 2018) uses support vector machines to predict relevant emotions and rankings in one text based on their intensities.
- **INN-RER** (Yang et al., 2018) designs a three-layer network combined with topic model to solve relevant emotion ranking.

Relevant emotion ranking can be treated as an extension of multi-label learning, so we also com-

Corpus	Category	Method	Criteria								
			PL(↓)	HL(↓)	RL(↓)	OE(↓)	AP(↑)	Cov(↓)	F1(↑)	MiF1(↑)	MaF1(↑)
News	Baselines	IRER-EA(-EA)	0.1786	0.1981	0.1023	0.0854	0.9122	1.9294	0.7423	0.7299	0.6116
		IRER-EA(-DEA)	0.1700	**0.1947**	0.1092	**0.0835**	0.9086	2.0014	0.7437	0.7358	0.6038
	Our model	**IRER-EA**	**0.1674**	0.1958	**0.0989**	0.0846	**0.9137**	**1.9108**	**0.7475**	**0.7379**	**0.6358**
Blogs	Baselines	IRER-EA(-EA)	0.2860	0.3010	0.1819	0.2004	0.7816	3.8320	0.5534	0.5422	0.3805
		IRER-EA(-DEA)	0.2852	0.3077	0.1844	0.1986	0.7936	3.8290	0.6027	0.6053	0.4220
	Our model	**IRER-EA**	**0.2609**	**0.2874**	**0.1638**	**0.1845**	**0.8149**	**3.6510**	**0.6304**	**0.6320**	**0.4804**
SemEval	Baselines	IRER-EA(-EA)	0.2760	**0.2968**	0.2076	0.1916	0.8565	**2.8235**	0.6494	0.6491	0.5480
		IRER-EA(-DEA)	0.2879	0.3220	0.2205	0.1534	0.8795	3.2893	0.6852	0.6891	0.5927
	Our model	**IRER-EA**	**0.2754**	0.3065	**0.1999**	**0.1286**	**0.8976**	3.5799	**0.7394**	**0.7538**	**0.6841**

Table 4: Comparison of different IRER-EA components."↓" indicates "the smaller the better", while "↑" indicates "the larger the better". The best performance on each evaluation measure is highlighted by boldface.

pare the proposed IRER-EA with several widely-used multi-label learning methods.

- **LIFT** (Zhang, 2011) constructs features specific to each label.
- **Rank-SVM** (Zhang and Zhou, 2014) distinguishes relevant labels from irrelevant ones with large margin strategy.
- **MLLOC** (Huang and Zhou, 2012) exploits local emotion correlations in expression data.
- **BP-MLL** (Zhang and Zhou, 2006) employs a novel error function into back propagation algorithm to capture the characteristics of multi-label learning.

For the MLL methods, linear kernel is used in LIFT. Rank-SVM uses the RBF kernel with the width σ equals to 1.

Experimental results on three corpora are shown in Table 3. It can be summarized from the table that: (1) IRER-EA performs better than state-of-art emotion detection baselines on almost all evaluation metrics across three corpora, which obviously shows the effectiveness of incorporating event information to obtain event-driven attentions for relevant emotion ranking; (2) IRER-EA achieves remarkably better results than MLL methods. It further confirms the effectiveness of IRER-EA, which uses a deep learning architecture incorporating event-driven attention for better performance.

4.2 Model Analysis

To further validate the effectiveness of event-driven attention components, we compare IRER-EA with two sub-networks based on our architectures.

- **IRER-EA(-EA)**: removes event-driven attention from IRER-EA.
- **IRER-EA(-DEA)**: removes document-level event-driven attention from IRER-EA.

Experimental results on three corpora are shown in Table 4. It can be summarized from the table that: (1) All the sub-networks cannot compete with IRER-EA on three corpora, which indicates the corpus-level and document-level event information are effective for relevant emotion ranking task; (2) IRER-EA(-DEA) performs better than IRER-EA(-EA) on most of the evaluation metrics, which verifies the effectiveness of incorporating corpus-level event-driven attention; (3) IRER-EA achieves better results than IRER-EA(-DEA) on almost all the evaluation metrics which further proves the effectiveness of document-level event-driven attention.

4.3 Case Study of Interpretability

To further investigate whether the event-driven attention is able to capture the event-associated words in a given document and provide interpretable results, we compare the results of the attention mechanisms of the word-level attentions and event-driven attentions by visualizing the weights of words in the same documents as shown in Figure 4. As the document of the News corpus and Blogs corpus are too long, we manually simplified the texts for better visualization results and provided English translations of the texts. The words marked in red represent highly-weighted ones according to the event-driven attentions, while the words with blue underlines are the

Corpora	Texts	Emotions
News	古琴台附近一名女子**轻生溺亡**。从她走进水里到最终**溺亡**，持续时间将近两分钟。当时有 10 余人**围观**，竟**无人**下水**施救**。这种现象让我们**寒心**。 A woman **drowned** near Guqin Terrace. It took nearly two minutes from she stepping into the water to her final **death**. More than 10 people were **watching** her, but **no one** came to **rescue**. Such phenomenon **chills** us.	Angry Sad Shocked
Blog	**地震**过后两百个小时里，每一张前沿的照片，每一段**获救**的视频，每个爸爸妈妈发自心底的痛哭，都令我心痛。 During the two hundred hours after the **earthquake**, each photo and video about **rescue** process and the **crying** from the parents make my heart broken.	Sorrow Love
SemEval	teacher in hide after **attack** on islam stir **threat**	Fear Sad

Figure 4: Case Study of Interpretability on Three Corpora.

ones with higher attention weights according to the word-level attentions.

From the visualization results on an example News article, it can be observed that different from word-level attention that pays more attentions to emotion-associated words, such as '**chill**' which may only evoke the emotion "*Sad*", the event-driven attentions can find words indicating latent events in the document, such as '**drown**', '**death**', '**no one rescue**' which are all closely related to the event "*Suicide without rescue*', which may evoke emotions such as "*Angry*" and "*Shocked*". In an example Blog article, word-level attentions highlight emotion-associated words such as '**crying**' and '**broken**' which may evoke the emotion "*Sorrow*", while event-driven attentions focus on the event-related words such as '**earthquake**' and '**rescued**' representing the event "*Earthquake Relief*'. Finally, in an example from the SemEval corpus, we can see that the word-level attention mechanism only gives a higher attention weight to the word '**threat**' and ignores the word '**attack**', which is also an important indicator of the emotions "*Fear*" and "*Sad*". On the contrary, the event-driven attention mechanism highlights both '**attack**' and '**threat**', representing the event "*Terrorist attack*'.

In summary, we can observe from Figure 4 that: (1) Event-driven attention can capture words representing latent events in texts; (2) Compared with the word-level attention which is prone to attend on emotion-associated keywords, event-driven attention can find words representing one or more hidden events in a document, which can provide more explainable clues of which event triggers certain emotions; (3) Event-driven attention can achieve better performance especially in documents without any emotion-associated words.

5 Conclusion

In this paper, we have proposed an interpretable relevant emotion ranking model with event-driven attention. The event information is incorporated into a neural model through event-driven attentions which can provide clues of how the emotions are evoked with explainable results. Moreover, corpus-level event embeddings and document-level event distributions are incorporated respectively to consider event information comprehensively. Experimental results show that the proposed method performs better than the state-of-the-art emotion detection methods and multi-label learning methods.

Acknowledgments

We would like to thank anonymous reviewers for their valuable comments and helpful suggestions. This work was funded by the National Key Research and Development Program of China (2017YFB1002801), the National Natural Science Foundation of China (61772132), the Natural Science Foundation of Jiangsu Province of China (BK20161430) and Innovate UK (grant no. 103652).

References

Saima Aman and Stan Szpakowicz. 2007. Identifying expressions of emotion in text. *Lecture Notes in Computer Science*, 4629:196–205.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. 2014. Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):2928–2941.

Andrew Cotter, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. 2011. Better mini-batch algorithms via accelerated gradient methods. *Advances in Neural Information Processing Systems*, pages 1647–1655.

S Hochreiter and J Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Sheng Jun Huang and Zhi Hua Zhou. 2012. Multi-label learning by exploiting label correlations locally. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 949–955.

Jingsheng Lei, Yanghui Rao, Qing Li, Xiaojun Quan, and Liu Wenyin. 2014. Towards building a social emotion detection system for online news. *Future Generation Computer Systems*, 37:438–448.

Kevin Hsin-Yih Lin, Changhua Yang, and Hsin-Hsi Chen. 2008. Emotion classification of online news articles from the reader's perspective. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 220–226. IEEE Computer Society.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. 2008. An introduction to information retrieval. *Journal of the American Society for Information Science and Technology*, 43(3):824–825.

W. Alan Nicewander. 1988. Thirteen ways to look at the correlation coefficient. *American Statistician*, 42(1):59–66.

Nikolaos Pappas and Andrei Popescu-Belis. 2017. Multilingual hierarchical attention networks for document classification. *CoRR*, abs/1707.00896.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing*, pages 1532–1543.

Rosalind W Picard and Roalind Picard. 1997. *Affective computing*, volume 252. MIT press Cambridge.

Changqin Quan and Fuji Ren. 2010. Sentence emotion analysis and recognition based on emotion words using ren-cecps. *International Journal of Advanced Intelligence Paradigms*, 2(1):105–117.

Xiaojun Quan, Qifan Wang, Ying Zhang, Luo Si, and Liu Wenyin. 2015. Latent discriminative models for social emotion detection with emotional dependency. *ACM Trans. Inf. Syst.*, 34(1):2:1–2:19.

Yanghui Rao. 2016. Contextual sentiment topic model for adaptive social emotion classification. *IEEE Intelligent Systems*, 31(1):41–47.

Yanghui Rao, Qing Li, Xudong Mao, and Wenyin Liu. 2014. Sentiment topic models for social emotion mining. *Information Sciences*, 266(5):90–100.

Yanghui Rao, Xiaojun Quan, Liu Wenyin, Qing Li, and Mingliang Chen. 2012. Building word-emotion mapping dictionary for online news. In *SDAD 2012 The 1st International Workshop on Sentiment Discovery from Affective Data*, page 28.

M. Schuster and K.K. Paliwal. 2002. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Fabrizio Sebastiani. 2001. Machine learning in automated text categorization. *Acm Computing Surveys*, 34(1):1–47.

Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74. Association for Computational Linguistics.

Yichen Wang and Aditya Pal. 2015. Detecting emotions in social media: A constrained optimization approach. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 996–1002.

Yang Yang, Deyu Zhou, and Yulan He. 2018. An interpretable neural network with topical information for relevant emotion ranking. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3423–3432.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. pages 1480–1489.

Min Ling Zhang. 2011. Lift: multi-label learning with label-specific features. In *International Joint Conference on Artificial Intelligence*, pages 1609–1614.

Min Ling Zhang and Zhi Hua Zhou. 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge Data Engineering*, 18(10):1338–1351.

Min-Ling Zhang and Zhi-Hua Zhou. 2014. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837.

Yue Zhang, Qi Liu, and Linfeng Song. 2018. Sentence-state lstm for text representation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 317–327. Association for Computational Linguistics.

Deyu Zhou, Yang Yang, and Yulan He. 2018. Relevant emotion ranking from text constrained with emotion relationships. In *Meeting of the North American Chapter of the Association for Computation Linguistics*, pages 561–571.

Deyu Zhou, Xuan Zhang, Yin Zhou, Quan Zhao, and Xin Geng. 2016. Emotion distribution learning from texts. In *Conference on Empirical Methods in Natural Language Processing*, pages 638–647.